

WHAT IS CLAIMED IS:

1. A method for processing video data to produce an effect to occur at a future time, comprising:

implementing an application thread, an upload thread, a decoding thread, a render thread, and a presenter thread.

passing the video data to the application thread for creating the effect to be added to the video data, generating pre-decompressed video data from the video data, and determining parameters which describe the effect;

passing the pre-decompressed video data to the upload thread for uploading the pre-decompressed video data into video hardware;

passing the pre-decompressed video data to the decoding thread for decoding the pre-decompressed video data to produce decoded video data;

passing the decoded video data to the render thread rendering the effect in the decoded video data to produce output video data; and

passing the output video data to the presenter thread to present the output video data.

2. The method according to claim 1, comprising implementing a release thread for releasing resources utilized in decoding and rendering.

3. The method according to claim 1, wherein the pre-decompressed video data is uploaded into video hardware using a Bus Mastering process.

4. The method according to claim 1, further comprising issuing a snooping command to determine a timing of each thread implementation.

5. The method according to claim 1, wherein the application thread performs:
reading a sample of the video data;
allocating a sample object for the sample;
partially decoding the sample to produce the pre-decompressed video data; and
transferring the sample object to the upload thread.

6. The method according to claim 5, wherein the upload thread performs:
obtaining a video memory surface;
issuing a first snooping command; and
uploading the pre-decompressed video data into the video memory surface.

7. The method according to claim 6, wherein the pre-decompressed video data is uploaded into the video memory surface using a Bus Mastering process.

8. The method according to claim 6, wherein the decoder thread performs:
issuing a second snooping command;
obtaining a new video memory surface;
determining a status of the new video memory surface;
performing the decoding to produce the decoded video data in the new video memory surface; and

attaching the new video memory surface to the sample object.

9. The method according to claim 8, wherein the application thread further performs:
determining, in the application, effect parameters for the effect;
passing the effect parameters from the application to the render thread.

10. The method according to claim 9, wherein the output sample object is a proxy.

11. The method according to claim 9, wherein the render thread performs:
assigning a target memory surface to the output sample object;
rendering the effect; and
storing the rendered effect in the target memory surface.

12. The method according to claim 11, wherein the presenter thread performs:
placing the output sample object in a presenter queue; and
performing a presenter method to present the output sample object as the output video data.

13. A method for processing video data to produce an effect to occur at a future time,
comprising the steps of:

receiving the video data;
creating the effect;
generating pre-decompressed video data from the video data;

uploading the pre-decompressed video data into video hardware;
decoding the pre-decompressed video data to produce decoded video data;
determining parameters which describe the effect;
rendering the effect in the decoded video data to produce output video data; and
presenting the output video data.

14. The method according to claim 13, comprising releasing resources utilized in decoding and rendering.

15. The method according to claim 14, wherein the steps of creating the effect, generating pre-decompressed video, and determining parameters are performed by an application.

16. The method according to claim 15, wherein the application initiates a thread for each step performed.

17. The method according to claim 15, wherein the steps of uploading the pre-decompressed video, decoding the pre-decompressed video data, rendering the effect, and releasing resources are performed by a 3D-Server.

18. The method according to claim 17, wherein the 3D-Server initiates a thread for each step performed.

19. The method according to claim 17, wherein the application and 3D-Server operate in parallel.

20. The method according to claim 13, wherein the pre-decompressed video data is uploaded into video hardware using a Bus Mastering process.

21. The method according to claim 13, further comprising issuing a snooping command to determine a timing of each step.

22. The method according to claim 17, wherein generating pre-decompressed video data from the video data comprises:

reading a sample of the video data;

allocating a sample object for the sample;

partially decoding the sample to produce the pre-decompressed video data; and

transferring the sample object to the 3D-Server.

23. The method according to claim 22, wherein uploading the pre-decompressed video data into video hardware comprises:

obtaining a video memory surface;

issuing a first snooping command; and

uploading the pre-decompressed video data into the video memory surface.

24. The method according to claim 23, wherein the pre-decompressed video data is uploaded into the video memory surface using a Bus Mastering process.

25. The method according to claim 23, wherein decoding the pre-decompressed video data to produce decoded video data comprises:

issuing a second snooping command;

obtaining a new video memory surface;

determining a status of the new video memory surface;

performing the decoding to produce the decoded video data in the new video memory surface; and

attaching the new video memory surface to the sample object.

26. The method according to claim 25, further comprising:

determining, in the application, parameters for the effect;

passing the effect parameters from the application to the 3D-Server; and

receiving, in the application, an output sample object.

27. The method according to claim 26, wherein the output sample object is a proxy.

28. The method according to claim 26, wherein rendering the effect in the decoded video data to produce output video data comprises:

assigning a target memory surface to the output sample object;

rendering the effect; and

storing the rendered effect in the target memory surface.

29. The method according to claim 28, wherein outputting the output video data comprises:

placing the output sample object in a presenter queue; and

performing a presenter method to present the output sample object as the output video data.

30. A system for processing video data to produce an effect to occur at a future time, comprising:

means for implementing an application thread for creating the effect to be added to the video data, generating pre-decompressed video data from the video data, and determining parameters which describe the effect;

means for implementing an upload thread for uploading the pre-decompressed video data into video hardware;

means for implementing a decoding thread for decoding the pre-decompressed video data to produce decoded video data;

means for implementing a render thread rendering the effect in the decoded video data to produce output video data; and

means for implementing a presenter thread presenting the output video data.

31. The system according to claim 30, comprising means for implementing a release thread for releasing resources utilized in decoding and rendering.

32. The system according to claim 30, wherein the pre-decompressed video data is uploaded into video hardware using a Bus Mastering process.

33. The system according to claim 30, further comprising means for issuing a snooping command to determine a timing of each thread implementation.

34. The system according to claim 30, wherein the means for implementing the application thread comprises:

means for reading a sample of the video data;

means for allocating a sample object for the sample;

means for partially decoding the sample to produce the pre-decompressed video data;

and

means for transferring the sample object to the upload thread.

35. The system according to claim 34, wherein the means for implementing the upload thread comprises:

means for obtaining a video memory surface;

means for issuing a first snooping command; and

means for uploading the pre-decompressed video data into the video memory surface.

36. The system according to claim 35, wherein the pre-decompressed video data is uploaded into the video memory surface using a Bus Mastering process.

37. The system according to claim 35, wherein the means for implementing the decoder thread comprises:

- means for issuing a second snooping command;
- means for obtaining a new video memory surface;
- means for determining a status of the new video memory surface;
- means for performing the decoding to produce the decoded video data in the new video memory surface; and
- means for attaching the new video memory surface to the sample object.

38. The system according to claim 37, wherein the means for implementing the application thread further comprises:

- means for determining, in the application, effect parameters for the effect;
- means for passing the effect parameters from the application to the render thread.

39. The system according to claim 38, wherein the output sample object is a proxy.

40. The system according to claim 38, wherein the means for implementing the render thread comprises:

- means for assigning a target memory surface to the output sample object;
- means for rendering the effect; and
- means for storing the rendered effect in the target memory surface.

41. The system according to claim 40, wherein the means for implementing the presenter thread comprises:

means for placing the output sample object in a presenter queue; and

means for performing a presenter method to present the output sample object as the output video data.

42. A system for processing video data to produce an effect to occur at a future time, comprising the steps of:

means for receiving the video data;

means for creating the effect;

means for generating pre-decompressed video data from the video data;

means for uploading the pre-decompressed video data into video hardware;

means for decoding the pre-decompressed video data to produce decoded video data;

means for determining parameters which describe the effect;

means for rendering the effect in the decoded video data to produce output video data;

and

means for presenting the output video data.

43. The system according to claim 42, comprising means for releasing resources utilized in decoding and rendering.

44. The system according to claim 43, wherein the means for creating the effect, means for generating pre-decompressed video, and means for determining parameters are an application.

45. The system according to claim 44, wherein the application initiates a thread for each step performed.

46. The system according to claim 44, wherein the means for uploading the pre-decompressed video, means for decoding the pre-decompressed video data, means for rendering the effect, and means for releasing resources are a 3D-Server.

47. The system according to claim 46, wherein the 3D-Server initiates a thread for each step performed.

48. The system according to claim 46, wherein the application and 3D-Server operate in parallel.

49. The system according to claim 42, wherein the pre-decompressed video data is uploaded into video hardware using a Bus Mastering process.

50. The system according to claim 42, further comprising means for issuing a snooping command to determine a timing of each step.

51. The system according to claim 46, wherein the means for generating pre-decompressed video data from the video data comprises:

- means for reading a sample of the video data;
- means for allocating a sample object for the sample;
- means for partially decoding the sample to produce the pre-decompressed video data;

and

- means for transferring the sample object to the 3D-Server.

52. The system according to claim 51, wherein the means for uploading the pre-decompressed video data into video hardware comprises:

- means for obtaining a video memory surface;
- means for issuing a first snooping command; and
- means for uploading the pre-decompressed video data into the video memory surface.

53. The system according to claim 52, wherein the pre-decompressed video data is uploaded into the video memory surface using a Bus Mastering process.

54. The system according to claim 52, wherein the means for decoding the pre-decompressed video data to produce decoded video data comprises:

- means for issuing a second snooping command;
- means for obtaining a new video memory surface;
- means for determining a status of the new video memory surface;

means for performing the decoding to produce the decoded video data in the new video memory surface; and

means for attaching the new video memory surface to the sample object.

55. The system according to claim 54, further comprising:

means for determining, in the application, parameters for the effect;

means for passing the effect parameters from the application to the 3D-Server; and

means for receiving, in the application, an output sample object.

56. The system according to claim 55, wherein the output sample object is a proxy.

57. The system according to claim 55, wherein the means for rendering the effect in the decoded video data to produce output video data comprises:

means for assigning a target memory surface to the output sample object;

means for rendering the effect; and

means for storing the rendered effect in the target memory surface.

58. The system according to claim 57, wherein the means for outputting the output video data comprises:

means for placing the output sample object in a presenter queue; and

means for performing a presenter method to present the output sample object as the output video data.

59. A computer readable medium including instructions for causing a computer system to execute a method for processing video data to produce an effect to occur at a future time, comprising:

implementing an application thread for creating the effect to be added to the video data, generating pre-decompressed video data from the video data, and determining parameters which describe the effect;

implementing an upload thread for uploading the pre-decompressed video data into video hardware;

implementing a decoding thread for decoding the pre-decompressed video data to produce decoded video data;

implementing a render thread rendering the effect in the decoded video data to produce output video data; and

implementing a presenter thread presenting the output video data.

60. The computer readable medium according to claim 59, comprising implementing a release thread for releasing resources utilized in decoding and rendering.

61. The computer readable medium according to claim 59, wherein the pre-decompressed video data is uploaded into video hardware using a Bus Mastering process.

62. The computer readable medium according to claim 59, further comprising issuing a snooping command to determine a timing of each thread implementation.

63. The computer readable medium according to claim 59, wherein implementing the application thread comprises:

- reading a sample of the video data;
- allocating a sample object for the sample;
- partially decoding the sample to produce the pre-decompressed video data; and
- transferring the sample object to the upload thread.

64. The computer readable medium according to claim 63, wherein implementing the upload thread comprises:

- obtaining a video memory surface;
- issuing a first snooping command; and
- uploading the pre-decompressed video data into the video memory surface.

65. The computer readable medium according to claim 64, wherein the pre-decompressed video data is uploaded into the video memory surface using a Bus Mastering process.

66. The computer readable medium according to claim 64, wherein implementing the decoder thread comprises:

- issuing a second snooping command;
- obtaining a new video memory surface;
- determining a status of the new video memory surface;

performing the decoding to produce the decoded video data in the new video memory surface; and

attaching the new video memory surface to the sample object.

67. The computer readable medium according to claim 66, wherein implementing the application thread further comprises:

determining, in the application, effect parameters for the effect;

passing the effect parameters from the application to the render thread.

68. The computer readable medium according to claim 67, wherein the output sample object is a proxy.

69. The computer readable medium according to claim 67, wherein implementing the render thread comprises:

assigning a target memory surface to the output sample object;

rendering the effect; and

storing the rendered effect in the target memory surface.

70. The computer readable medium according to claim 69, wherein implementing the presenter thread comprises:

placing the output sample object in a presenter queue; and

performing a presenter method to present the output sample object as the output video data.

71. A computer readable medium including instructions for causing a computer system to execute a method for processing video data to produce an effect to occur at a future time, comprising the steps of:

receiving the video data;

creating the effect;

generating pre-decompressed video data from the video data;

uploading the pre-decompressed video data into video hardware;

decoding the pre-decompressed video data to produce decoded video data;

determining parameters which describe the effect;

rendering the effect in the decoded video data to produce output video data; and

presenting the output video data.

72. The computer readable medium according to claim 71, comprising releasing resources utilized in decoding and rendering.

73. The computer readable medium according to claim 72, wherein the steps of creating the effect, generating pre-decompressed video, and determining parameters are performed by an application.

74. The computer readable medium according to claim 73, wherein the application initiates a thread for each step performed.

75. The computer readable medium according to claim 73, wherein the steps of uploading the pre-decompressed video, decoding the pre-decompressed video data, rendering the effect, and releasing resources are performed by a 3D-Server.

76. The computer readable medium according to claim 75, wherein the 3D-Server initiates a thread for each step performed.

77. The computer readable medium according to claim 75, wherein the application and 3D-Server operate in parallel.

78. The computer readable medium according to claim 71, wherein the pre-decompressed video data is uploaded into video hardware using a Bus Mastering process.

79. The computer readable medium according to claim 71, further comprising issuing a snooping command to determine a timing of each step.

80. The computer readable medium according to claim 75, wherein generating pre-decompressed video data from the video data comprises:

reading a sample of the video data;

allocating a sample object for the sample;

partially decoding the sample to produce the pre-decompressed video data; and

transferring the sample object to the 3D-Server.

81. The computer readable medium according to claim 80, wherein uploading the pre-decompressed video data into video hardware comprises:

- obtaining a video memory surface;
- issuing a first snooping command; and
- uploading the pre-decompressed video data into the video memory surface.

82. The computer readable medium according to claim 81, wherein the pre-decompressed video data is uploaded into the video memory surface using a Bus Mastering process.

83. The computer readable medium according to claim 81, wherein decoding the pre-decompressed video data to produce decoded video data comprises:

- issuing a second snooping command;
- obtaining a new video memory surface;
- determining a status of the new video memory surface;
- performing the decoding to produce the decoded video data in the new video memory surface; and
- attaching the new video memory surface to the sample object.

84. The computer readable medium according to claim 83, further comprising:

- determining, in the application, parameters for the effect;
- passing the effect parameters from the application to the 3D-Server; and
- receiving, in the application, an output sample object.

85. The computer readable medium according to claim 84, wherein the output sample object is a proxy.

86. The computer readable medium according to claim 84, wherein rendering the effect in the decoded video data to produce output video data comprises:

assigning a target memory surface to the output sample object;

rendering the effect; and

storing the rendered effect in the target memory surface.

87. The computer readable medium according to claim 86, wherein outputting the output video data comprises:

placing the output sample object in a presenter queue; and

performing a presenter method to present the output sample object as the output video data.